

## **REMARKS**

The Examiner rejected claims 1-9 and 19-22 under §112, paragraph 1, for failing to comply with the enablement requirement. The Examiner further rejected claims 24 and 25 under §112, paragraph 2, for indefiniteness. In addition, the Examiner rejected claims 10-13, 23, and 24 under §102 as being anticipated by U.S. Patent 6,470,388 issued to Niemi et al. ("Niemi"). The Examiner rejected claims 1-3, 11, 14-16, 19 and 21 under §103 as being unpatentable over Niemi in view of U.S. Patent 6,144,967 issued to Nock ("Nock"). The Examiner rejected claims 17, 18, 20, 22, 25 under § 103 as being unpatentable over Niemi in view of Nock and U.S. Patent 5,740,354 issued to Ben-Natan et al. ("Ben-Natan"). The Examiner rejected claim 4 under §103 as being unpatentable over Niemi in view of Nock and U.S. Patent 5,819,094 issued to Sato et al. ("Sato").

In this Amendment, Applicants have amended claims 1-2, 4-9 and 11-25. Additionally, Applicants have added new claims 26-38. No claims have been canceled. Accordingly, claims 1-38 will be pending after entry of this Amendment.

### **I. Enablement Rejections Under § 112, Paragraph 1**

The Examiner rejected claims 1-9 and 19-22 under §112, paragraph 1, for failing to comply with the enablement requirement. The Examiner stated that while claims 1-9 and 19-22 recite an event object occupying a memory space and executing independent of said applications, there was no description of executing event objects in the Specification.

The objected claim language of claims 1, 19 and 21 was a part of these claims as they were originally filed. As such, this language was part of the Specification of the application at the time of filing. However, to obviate this issue, Applicants have amended claims 1 and 19 to remove reference to execution of any object.

The Examiner also rejected claim 23 under §112, paragraph 1 for failing to provide enabling disclosure for the use of the phrase “a third event-logging application.” The Examiner stated that the Specification referred to an event-logging mechanism and not an event-logging application. Applicants respectfully submit that as commonly understood in the art, the term application refers to a set of code that performs a set of operations. As such, Applicants believe that the use of the term application was appropriate in the prior claim 23. However, to obviate this issue, Applicants have replaced the term “application” with the term “mechanism,” which was cited by the Examiner. In view of these amendments, Applicants respectfully request reconsideration and withdrawal of the §112, paragraph 1 rejections.

## **II. Indefiniteness Rejections Under § 112, Paragraph 2**

The Examiner rejected claims 24 and 25 under §112, paragraph 2, for lack of enablement with respect to an event logging mechanism being an application and for indefiniteness because the phrases “the event object” and “said third event-logging mechanism” have improper antecedent basis. In this Amendment, Applicants amended the phrase “the event object” in claim 24 to “an event

object.” Applicants have also amended “third event-logging application” to “third event-logging mechanism” in claim 23, which addresses the antecedent basis issue cited by the Examiner for claim 25. In view of these amendments, Applicants respectfully request reconsideration and withdrawal of the §112, paragraph 2 rejections.

### **III. Rejection of Claims 1-4, and 19-22 Under § 103**

The Examiner rejected claims 1-3 under § 103 as being unpatentable over Niemi in view of Nock. The Examiner rejected claim 4 under § 103 as being unpatentable over Niemi in view of Nock and Sato. The Examiner rejected claim 19 under §103 as being unpatentable over Niemi in view of Nock. The Examiner rejected claim 20 under §103 as being unpatentable over Niemi in view of Nock and Ben-Natan. The Examiner rejected claim 21 under §103 as being unpatentable over Niemi in view of Nock. The Examiner rejected claim 22 under §103 as being unpatentable over Niemi in view of Nock and Ben-Natan. The Examiner asserts that Niemi teaches event logging that logs start time and gets debug objects or states without involving actions of an applications program and that Nock logs an end time.

Claims 2-4 are dependent directly or indirectly on claim 1. Claim 1 recites a method that creates an event object for an event to be logged that has not yet been logged within an application. An event object so created, occupies a memory space that is independent of the application. The method also logs within the event object the start time, end time and information regarding the event.

For two reasons, Applicants respectfully submit that the cited references fail to disclose, teach, or even suggest such a method. First, the § 103 rejection of claim 1-4 relies on impermissible piecemeal, hindsight combination of the references. These rejections are devoid of any suggestion or motivation in the art for the piecemeal, hindsight combination of the references.

Second, even the piecemeal, hindsight combination of references does not disclose an event logging method that creates an event object for an event to be logged that has not yet been logged within an application. Niemi teaches a method of logging event data in a network that generally includes a distributed processing environment. *Niemi, Column 5, lines 21-26*. The distributed processing system has a central logger and multiple nodes. Each node has a local logger. *Niemi, Column 5, lines 57-63*. The application provides logging at the local node regardless of whether logging is on or off. If logging is enabled, a local event object message packet is passed to the central logger to be parsed and stored in a central location. *Niemi, Column 11, lines 53-59*. At the central logger, a new record 514 is created based upon the information received from the distributed node about a previously logged event. *Niemi, Column 11, line 66 through Column 12, line 15*. The cited record 514 logs events that have already been logged by a debug object in the application 208a/208b (See Niemi, columns 11 and 12). Because the Niemi central logger can only create data from the event logs of the local logger, Niemi does not disclose, teach, or even suggest creating an event object for an event to be logged that has not yet been logged within an application.

The subject claims were also rejected because “Nock teaches for each event, logging an end

time (end section event) in addition to a start time (begin section event). Column 13, line 9 – column 14, line 13.” However, Nock is a log analysis system (*Column 12, lines 56-58*) that receives its data from a previously created event log. *Nock, Column 13, lines 6-8*. It solely reviews data that was previously logged. *Nock, Column 12, lines 56-58*. To start the analysis, Nock retrieves a previously written event log and looks for a logical start-of-function event. *Id at Column 13, lines 1-3, Column 13, lines 26-50*. After finding one, it places a start-of-event mark in its copy of the log and looks for a corresponding end-of-event. *Id*. When it finds the corresponding end of event, it places a stop event mark in its copy of the log. *Id at lines 51-62*. This provides the analysis tool with a section of data having a logical start and end. No time stamp is placed on either the start or the stop mark of the analyzer’s copy of the event log at any time, including at the completion of the event.

Therefore, Applicants respectfully submit that claim 1 is not rendered unpatentable by the piecemeal, hindsight combination of Niemi, Nock, and/or Sato. As claims 2-4 are dependent on claim 1, the same argument applies to those claims as discussed. In view of the foregoing, Applicants request reconsideration and withdrawal of § 103 rejection of claim 1-4.

Claims 19 and 21 are similar to claim 1, except that claim 19 recites a computer readable medium, while claim 21 recites means plus function limitations. Claim 20 is dependent on claim 19, while claim 22 is dependent on claim 21. Applicants respectfully submit that claims 19-22 are patentable over the piecemeal, hindsight combination of references used for rejecting these claims under § 103. In view of the foregoing, Applicants request reconsideration and withdrawal of § 103

rejection of claim 19-22.

#### **IV. Rejection of Claims 10-18 Under §§ 102 and 103**

The Examiner rejected claims 10 and 13 under §102 as being anticipated by Niemi. The Examiner rejected claims 11, 12, and 14-16 §103 as being unpatentable over Niemi in view of Nock. The Examiner rejected claims 17 and 18 under §103 as being unpatentable over Niemi in view of Nock and Ben-Natan.

Claims 11-18 are dependent upon claim 10. Claim 10 recites a system that comprises a foundational layer upon which applications are built and executed. The system further comprises an event logging mechanism operating independently of the applications. This event logging mechanism generates an event log for any of the applications. The event logging mechanism operates without referencing any event logs of the applications.

Applicants submit that the cited references do not disclose, teach, or even suggest an event logging system that operates without referencing any event logs of the applications. Niemi discloses an event logging system in a distributed processing environment wherein each node of the network has a local event logger that will report events to a central event logger if logging is enabled. *Niemi, column 5, line 57 through column 6, line 10*. Before that happens, a debug object in each instance is created by the application (*Niemi, column 9, lines 60-65*) and the local event logger notes and logs the occurrence of an event and includes information about the event, such as the debug object

information. *Id. at column 6, lines 10-19.* If logging is enabled when an event occurs, the central logger is notified and receives a copy of the local event log. *Id at column 11 – 12.*

The production of record 514, created from information in the local event log, includes retrieving and copying event logs of the application. Record 514 is a new record or data entry corresponding to a received Log() service request. *Niemi, column 12, lines 10-11.* The Log() service request is encapsulated in one or more data packets and transmitted from the local logger to the central logging facility. *Niemi, column 11, lines 59-67.* The data packets include a message field. *Niemi, column 11, line 59.* The message field contains “the actual trace, audit or other information generated by the corresponding application or process.” *Niemi, column 12, lines 30-33.*

Thus, record 514 contains data that is initially generated by the application and later accessed by the central logger and it is incorrect to assert that Niemi teaches creating event log data without any reference to event logs of the application. Accordingly, Applicants respectfully submit that claim 10 is not anticipated by Niemi. As claims 11-18 are dependent on claim 10, the same argument applies to the piecemeal, hindsight § 103 rejection of claims 11-18. In view of the foregoing, Applicants request reconsideration and withdrawal of §§ 102 and 103 rejections of claim 10-18.

## **V. Claims 23-25**

The Examiner rejected claims 23 and 24 as being anticipated under §102 by Niemi. The

Examiner also rejected claim 25 as being obvious under § 103 over the combination of Niemi, Nock and Ben-Natan.

Claims 24 and 25 depend on claim 23. Claim 23 recites a system that comprises a foundational layer upon which applications are built and executed and a first and second application for executing on the foundational layer. The system further comprises an event-logging center for execution on the foundational layer. This event-logging center functions interoperably with but separately from the first and second applications. The event-logging center generates an event log for each of the first and second applications, where at least one of the first or second applications does not generate an event log.

Applicants submit that the cited references do not disclose, teach, or even suggest such a system. Specifically, the cited references do not disclose, teach, or even suggest generating an event log for each of the applications, where at least one of the applications is not generating its own event log.

As noted above, Niemi comprises a dual logging system operating in a distributed network environment. One logging system is located at each local node and logs events for applications executing on that node. There may be many local logging systems. The other logging system is a central logger. Within the network, there is only one central logger. The local logging systems function regardless of the enabled/disabled state of the central logger. When enabled, the local



loggers notify the central logging system of an event and pass to the central logger information packets which include data regarding the event. Thus, the central logger does not log events that are not being logged by either a first or a second application.

As noted above in detail, Nock does not teach placing the end-of-event time in the object log (either when the event completes or at any other time). Ben-Natan also fails to teach these claim limitations. Thus, Applicants respectfully submit that claim 23 is not anticipated by Niemi. As claims 24 and 25 are dependent on claim 23, the same argument applies to the §§ 102 and 103 rejections of claims 24 and 25. In view of the foregoing, Applicants request reconsideration and withdrawal of §§ 102 and 103 rejections of claim 23-25.

#### **VI. Claims 26 – 38**


This Amendment adds claims 26 – 38. Applicants respectfully submit that these claims are patentable over the cited references.

## CONCLUSION

In view of the foregoing, it is submitted that all pending claims, namely claims 1-38, are in condition for allowance. Allowance is earnestly solicited at the earliest possible date.

Respectfully submitted,  
STATTLER, JOHANSEN & ADELI LLP

Dated: 7/28/15

  
\_\_\_\_\_  
David L. Bell  
Reg. No. 45,444

Stattler, Johansen & Adeli LLP  
1875 Century Park East, Suite 1050  
Los Angeles, CA 90067  
Phone: (310) 785-0140 x301  
Fax: (310) 785-9558